Metrics Are Needed for Collaborative Software Development

Mojgan Mohtashami
Advanced Infrastructure Design
10 Richardson Lane
Hightstown, NJ 08520
mojgan@aidpe.com

Cyril S. Ku
Department of Computer Science
William Paterson University
Wayne NJ 07470
kuc@wpunj.edu

Thomas J. Marlowe
Department of Mathematics and
Computer Science
Seton Hall University
South Orange NJ 07079
thomas.marlowe@shu.edu

Abstract

There is a need for metrics for inter-organizational collaborative software development projects, encompassing management and technical concerns. In particular, metrics are needed that are aimed at the collaborative aspect itself, such as readiness for collaboration, the quality and/or the costs and benefits of collaboration in a specific ongoing project. We suggest questions and directions for such metrics, spanning the full lifespan of a collaborative project, from considering the suitability of collaboration through evaluating ongoing projects to final evaluation of the collaboration.

Keywords: Collaboration, metrics, software engineering, ICSD, management contingency policies, risk management, intellectual property.

1 INTRODUCTION

Software engineering is essentially collaborative for any but the smallest and simplest projects. But the extent of collaboration has increased from multiple teams at a single site, to distributed teams in a single organization, to collaborating teams in multiple organizations. Yet not all collaborations are successes. As collaboration broadens, and as projects become more complex and long-lived, it becomes ever more important to have management oversight, technical coordination and supervision, and quality control. But once the projects become too large for day-to-day personal contact, these command, control, coordination and communication (4-C) factors need to rely more and more on reports and metrics [1,8,9,21,24]—and, indeed, metrics have been created for every phase of software engineering and its management, for project, process and product, for the single-team, multi-team, and distributed models of development [2]. There are also metrics for knowledge management [22,23,24] and for communication [16], which assume far greater importance in a complex, long-lived and evolvable software development project [4,5,6,15]. However, there is little in the literature on metrics specifically devoted to collaboration.

Inter-organizational Collaborative Software Development (ICSD) [6,19,20,21] is here understood as: multiple institutions working together on complex, long-lived, evolving software. Each institution is responsible for one or more product components, product aspects, process activities, or business tasks, and neither institutional nor product roles and responsibilities are completely constrained by initial agreement. Collaboration affects every software development phase and activity [11]. The obvious complications occur not only in the usual product and process incompatibilities, but also with intellectual property, privacy and security, on the one hand, and knowledge management and risk analysis, on the other.

ICSD has multiple implications, some of which are fairly subtle. All aspects of development and its context need to become even more modular and hierarchical, typically entailing parallel partner and collaboration structures paired with conflict resolution mechanisms. New risks are introduced at the same time that risk management becomes more complex [17.18]. Interfaces need to be more highly specified yet retain flexibility. Moreover, granularity of knowledge becomes an inherent issue, not just a convenience for efficiency or abstraction, but isolation of the internals of one partner’s components and processes to address intellectual property, privacy and security concerns.

As ICSD becomes more common, metrics need to be revised or created to support this mode of development. There are two major dimensions. First, as we discussed in [12], creating or modifying metrics for the standard criteria for software project, process and product: for example, measures of effort and time, of software quality, of process compliance, and of test coverage. Second, metrics aimed at the collaborative aspect itself, such as readiness for collaboration, the quality and or the costs and benefits of collaboration in a specific ongoing project.

Example metrics of the first type include product metrics focused on structural complexity (at specification time, during design, or on release) or test coverage; process metrics aimed at process compliance or quality assurance; and project metrics for schedule, budget, staffing and/or training; metrics on the effectiveness of risk management.

Such metrics typically comprise a selection of a set of key performance indicators, and a weighted combination to a balanced scorecard. The identification of suitable factors, and the weights to be used, relies on results from knowledge management, statistical factor analysis, and domain processes and practices from management science, software engineering, risk management, and requirements analysis.

In this submission, we concentrate on the second facet. We consider what metrics are needed, some difficulties, and some issues. We divide our metrics in two dimensions—staging (when are the metrics useful) and focus (corporate, infrastructure, people, project, process, product). It will of course also be useful to adjust the metrics to the type of collaboration anticipated—whether, for example, largely separate design of components for later integration or coordinated or complementary use (strategic collaboration), or coordinated development of a single product or product suite, interacting in every software engineering activity (tactical collaboration), or an intermediate form. We intend to address this issue, as well as propose metrics, in future work.

2 PRE-COLLABORATION METRICS

These divide into generic (ready for collaboration?) and specific (ready for this collaboration?) metrics. Generic metrics should consider the following issues [16].

- Corporate: Are we willing and ready to participate in a collaborative software development venture? A reasonable metric will combine survey data with a checklist of criteria. A survey of key personnel can establish support among management, IT department heads, technical managers, and so on, as well as attitudes toward crediting employees for success in collaborative ventures. The checklist should include the degree to which policies, procedures and practices favor collaboration: in particular, intellectual property and information sharing, risk management, and knowledge management. To the extent possible, a similar checklist should be applied for evaluation of (proposed) partners.
- Infrastructure and technical: Is there a robust, multi-mode communication infrastructure in place? Are processes, practices and tools amenable to collaboration? Which CMMI maturity level [3] is in-place in the organization, and has this been adapted for collaboration?
  What is the state of (proposed) partner infrastructure and policies? Do (proposed) partners have the requisite expertise at the required level? Do we have a common (or interconvertible) set of processes, conventions and notations, and a common glossary?
- People (human resources): Is there appropriate support for training and cultural sensitivity? Are there corporate and technical managers who will be willing and able to work with counterparts in other organizations? Are there obstacles to collaboration in the people or policies in corporate management, or in the legal or IT departments [17,18]?

Metrics aimed at specific projects and products should attempt to determine the appropriateness of the project and product for collaborative development, and (if possible) the appropriateness and quality of the proposed collaborators. The latter will definitely include evaluation of past relationships with other collaborators or their key personnel. It is also of course important that the project and product have clear and viable objectives, be a good strategic and tactical fit with institutional vision and mission, and have good agreement with partner experience and expertise.

Technical considerations include the following.

- How natural is the decomposition of this project into components? Are the boundaries relatively clear? Does the component decomposition fit with the expertise of the proposed partners?
- To the extent that components or interfaces are fuzzy, does the software process allow for any flexibility in interfaces?
- To the extent that innovation, novel interfaces, or use of scientific or technical information is part of the product, is there a provision for conferencing and meetings of domain or discipline experts?

In addition, we must be able to measure the willingness of the partners to establish required structures.

- Are the partners, and the collaboration as a whole, willing to create, maintain and support a shared technical infrastructure, including communication media and protocols, electronic and in-person meetings and consultations, shared tools and views, knowledge management, and risk management?
- Are the partners, and the collaboration as a whole, willing to create, maintain and support a management superstructure, both in the individual partners and collaboratively, to provide direction, support and championship?
- Will the collaborative agreement provide for reflection and evolution in collaborative structures and processes, and do such processes exist for individual partner structures, policies and processes?
- Are the partners, and the collaboration as a whole, willing to create, maintain and support methods for resolving ambiguities, conflicts and difficulties, whether technical, corporate, or legal?
- Is there a clear strategy and allocation of responsibility for marketing (or using) the product? Is there a clear allocation of responsibilities for maintenance and evolution (or a process for determining these)?

The following are templates for two useful pre-collaboration metrics. The values and weightings, and the method for assigning those are subjects of future work.

2.1 BUSINESS INFRASTRUCTURE READINESS

Business infrastructure readiness must consider the suitability of the communication and development platforms, and the existence of shared/sharable artifacts. The evaluation needs to be carried out by each partner, for themselves, their prospective partners, and the collaboration as a whole.

A robust communication platform should support both synchronous and asynchronous modes, formal and informal electronic communication and interpersonal communication; and support, repositories, and configuration management for documents, messages, and artifacts, with virtual meeting and multi-user editing facilities. Virtual meetings and other synchronous communication must adapt to geographical and particularly temporal dispersion of participants.

All policies and practices should be collaboration-aware (C-A), and in particular IT policies and procedures, including firewalls, should support and not interfere with collaboration. Likewise, software engineering, risk management, and knowledge management policies, practices and tools should be amenable to collaboration. Finally, the appropriate domain-specific certifications and practices should have been achieved—or should be possessed by potential partners.

In addition, in evaluation of the collaboration, the existence of shared glossaries, notations and conventions—for risk and knowledge management as well as for software engineering, and familiarity with them, or with artifacts that can be transformed into them, should be established.

2.2 PROJECT SOFTWARE STRUCTURE

Once the partnership is formed, the decomposition into partner responsibilities is of primary importance. Evaluation of the quality of this decomposition is crucial to determining whether to proceed with the venture. There are at least three major issues.

First, is the decomposition into components clear and natural, and a good fit with partner expertise and capabilities? Second, are the boundaries and interface specifications clear, or if fuzzy, is there flexibility in interface definition, and does this flexibility correspond to and address the perceived lack of precision?

Finally, and particularly if the project or product requires innovation, novel interfaces, of heavy use of scientific or technical information, are the project and process flexible enough to accommodate resulting pressures? In particular, does the project budget and schedule allow for flexibility in goals and in partner responsibilities? And is there provision for conferencing and/or meetings of domain or discipline experts?

Table 1. Overview of Pre-Collaboration Metrics

	Corporate/ Infrastructure	Technical/ Development	Knowledge/ Risk
Collaboration	Advocacy C-A policies	Communication C-A processes	Maturity C-A processes
This project	Partners Shared artifacts	Decomposition Interfaces	Intell. Property
Infrastructure	IT support	Tools Training	KBs Security policy

3 METRICS FOR ONGOING COLLABORATION

In addition to the standard (if modified) metrics, it will be important to have several other classes of metrics.

The first would measure the quality of the ongoing collaboration, complementing schedule and cost tracking with evaluation of the success, use and usefulness, and problems of the collaborative structures. These include measurements of the clarity of interface specification (have problems arisen? Are they due to differences in language or culture?), risk management (have unanticipated collaborative risks emerged?), management cooperation, or problems with infrastructure? This would be itself complemented by ongoing measurements, assuring that the project and product continued to fit with strategic and mission objectives, and that collaborative and partner support structures were continuing to act and to function as required.

The second, interacting with risk management and knowledge management, aims at early detection of problems—which of course interacts with the first. These may arise from corporate, legal, or people issues in the collaboration, or from stresses and changes in the development process resulting from collaboration. In [7,10,13,14,17,18], we have identified a number of these stresses, and proposed a number of changes to project and process artifacts to support collaboration, good software engineering, and evolvable systems. It should be noted that significant stresses include the quality of both structures and processes for partner and collaborative risk management and knowledge management.

Third, in long-lived and knowledge-intensive projects, it will be necessary during the current project, and for the development and maintenance of trust in ongoing relationships, to be able to assign credit for knowledge and services provided by one partner to another, or to the collaboration as a whole, and costs for the use of others' knowledge and services. This is important in particular where long-lived projects need to maintain organizational partners in the face of a turnover of most or all of the original team members. This assignment is complicated by cases in which the product, the integration of components, the collaborative process itself, or the analysis of any of these, generates knowledge, but needs to accommodate that possibility. Even approximate measures of value will allow such metrics to be maintained.

Finally, it would be helpful, both during the collaboration and in post-project evaluation, to have a metric of the costs and benefits of collaboration, ideally a fine-grained metric so that different areas and forms of collaboration could be evaluated. During the collaborative process, such metrics could focus attention on problem areas, and perhaps indicate areas in which the task decomposition could be revisited. This would also have substantial benefits for future collaborations on similar projects, perhaps suggesting facets or attributes that might be best left in control of a single partner, or as an input on the decision to collaborate or use single-developer mode.

Also note that the quality and utility of these metrics depends both on their timeliness and the quality of data collected. Thus, in addition, a process and accompanying metrics will be needed to assure timely, consistent, and accurate data from each organization, and where relevant, from collaborative structures.

As usual, this process will benefit from common or compatible approaches and tools for data gathering, storage and communication, data quality assessment, and so on.

3.1 COLLABORATIVE COST-BENEFIT ANALYSIS

Ongoing evaluation of the utility of collaboration entails the standard examination of risks, costs, benefits and opportunities. The obvious benefit is the time and money saved vis-à-vis a hypothetical single-organization or contractual development; less obvious benefits include the risks, errors, and flaws avoided, plus unexpected enhancements, optimizations, and preferred implementations, as well as the knowledge and expertise acquired, generated, obtained and maintained—although this must be balanced against lost opportunities to obtain such expertise by developing the other components. The standard examination approach can also be seen as embedded within normal procedures, reducing the risk of resistance within the ranks of processweary personnel.

Costs including the start-up costs in preparing for and initiating the collaboration—although these are amortized over the history of collaboration and interaction with the same partners. Ongoing costs arise from difficulties encountered in working with unsatisfactory partners, handling interface errors and incompatibilities, as well as addressing other collaborative risks, and the overhead of maintaining collaborative structures and infrastructure, including collaborative risk management and knowledge management.

Finally, risks include loss of relationships with customers and users, and with partners, as a result of a failed collaboration, an unsatisfactory partner, or poor handling of interorganizational interactions.

Conducting such an evaluation on an ongoing basis has impacts on both the current and future projects. For the current project, it presents opportunities to quickly identify and focus on problem areas, to identify problems with decomposition, interfaces or responsibilities, to evaluate collaborative structures and relationships, and to provide feedback for risk management and possibly for knowledge management as well.

Impacts on future projects include tuning the collaborative readiness metric and collaborative structures, improving the “Collaborate-Contract-Work Alone” decision, optimizing project decomposition and responsibilities, and assisting in identifying good and unsatisfactory partners for future collaborations.

4 POST-COLLABORATION METRICS

One class of post-project metrics will mirror pre-collaboration and/or mid-collaboration metrics. How well did a particular corporate facet or collaborative function perform (communication support, intellectual property control, risk management, etc.)? How well did the collaboration function, and what problems need to be addressed? Did project management and software process function as expected, or what should be changed?

A second class deals with the overall success of the project and the collaboration itself. The real questions that need to be answered are: Was the project a success? Did the product meet its functional and non-functional requirements? Did the project, process and product meet quality targets? Did collaboration help or hinder in meeting schedule and budget? And did the project and the product fulfill partner and collaborative business objectives?

If it was not a success, was the project worth trying? Was the collaboration a success? How did collaboration affect the success of the project? One tricky point is that some projects would never have been undertaken by any of the partners acting alone. (Since many if not most software development projects do not succeed fully, it is not clear that partial failure of a collaborative venture is in and of itself evidence that collaboration is not viable.)

Finally, what have we learned? What changes are needed—in structures, artifacts, staging, or management? Are revisions needed in collaborative configuration and change management, risk management, knowledge management, or metrics and quality assurance themselves? We consider one post-collaboration metric below.

4.1 EFFECTIVENESS OF COLLABORATIVE KNOWLEDGE MANAGEMENT

How many queries to the collaborative knowledge base required access to information from multiple partners (both as an absolute measure and as a percentage of total queries)? From the collaborative knowledge base? From analysis of multiple components of the product?

What proportion of the information in the collaborative knowledge base resulted from integration? From analysis? Required both?

How much credit would be assigned to the collaboration as a whole if it were considered a partner? How much collaborative information is needed to analyze, test, maintain, or modify individual partner components? To modify interfaces or collaborative structures and practices?

On the cost side: How often did inference or representation of information fail because the abstraction or filtering hid required information? That is, in principle, when it would have succeeded if this had been a single-organization project?

How much effort, time and money were used in determining representations for the collaborative knowledge base, or encoding/decoding its information? How often was human intervention required for the encoding?

These questions also assume that there is an awareness of such issues, with the experience to understand the value and deliver crucial information from collaborative knowledge embedded within organizational frameworks. Thus, the cost of training and sensitization, and of development of algorithms and approaches for identifying integrative and collaborative knowledge [15] must also be taken into account, although it may be possible to amortize it across a set of collaborative projects..

5 CONCLUSIONS

We have argued a need for metrics for collaborative software development, and in particular metrics aimed at the collaboration itself, and have considered many of the important questions that will have to be addressed by such metrics. See Figure 1 for an overview. Many but not all of these issues apply to collaboration in general. Future work will entail

- Developing metrics, via interviews, surveys, and case histories, and applying these to collaborative software development projects.
- Applying our approach to a broader range of inter-organizational collaboration on other technical projects, particularly intellectual property ventures and those with complex collaborative structure, or a complex, long-lived and evolvable product.
- Integrating these metrics with new or revised versions of traditional metrics into a metric suite.
- Investigating interactions of these metrics, and their interaction with ongoing technical and business processes, with the aim of determining correlations as well as co-regulative negative and positive synergistic feedback/feedforward loops.

One difficulty is that corporate management may resist some forms of metrics, so that obtaining data may be a serious problem. Collaboration is likely to intensify this.

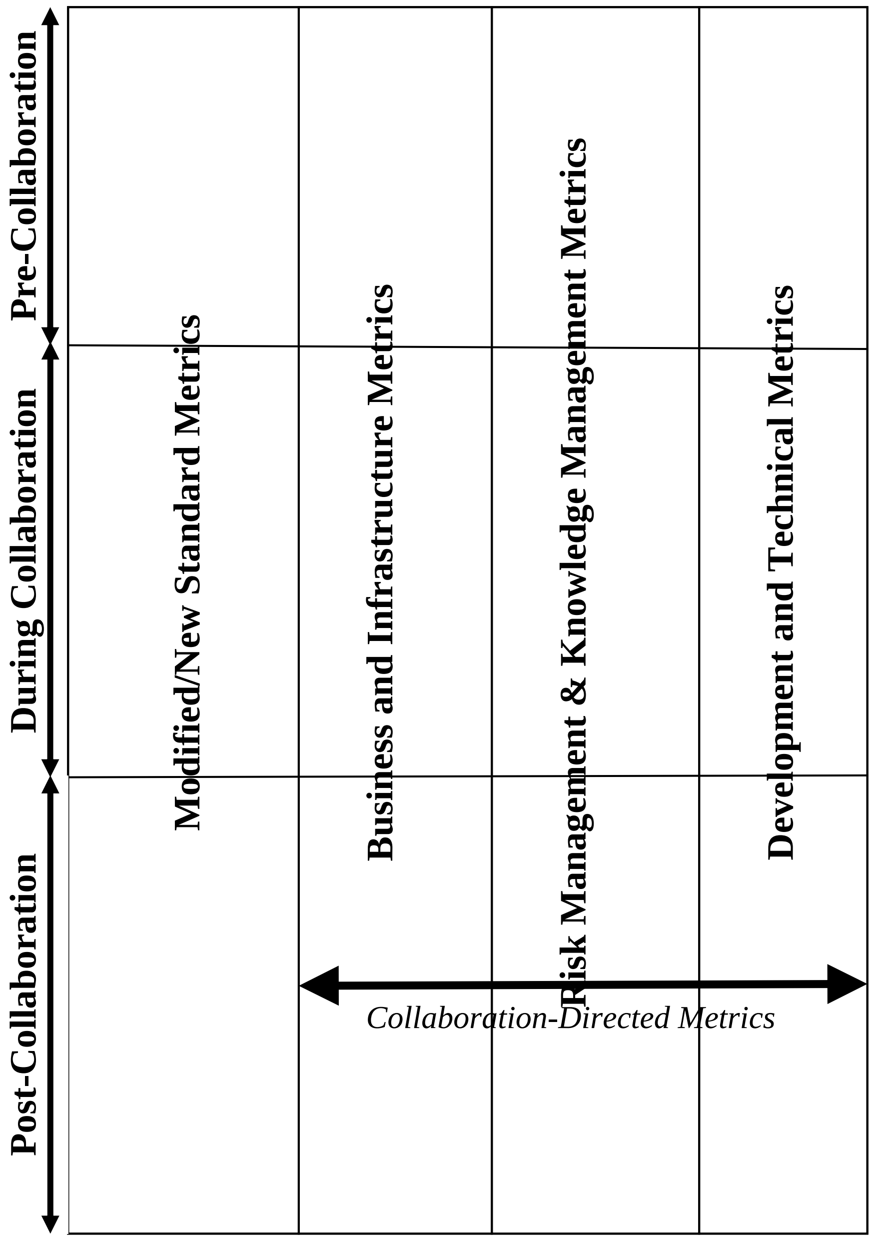
The same factors that resist collaboration are likely to resist sharing internal data that is apparently not needed for the functioning of the collaboration or the development of its product. Resistance will be intensified where there is little trust of the quality or validity of data supplied by the partners—both to providing the data, and to believing the results.

While increasing trust between partners addresses this question in part, trust alone is not sufficient. Thus future work—ours or others'—will also be needed to develop structures and processes to assure consistency of interpretation of measures, together with quality and timeliness of data.

In sum, development of metrics to measure collaboration, and adaptation of traditional metrics to a collaborative context, is a key step in developing a viable framework for successful collaboration.

.

Figure 2. Overview of Metrics for Inter-Organizational Collaboration



Acknowledgements

The authors wish to thank Norbert Jastroch, Vassilka Kirova, Susu Nousala, and the anonymous referees, for their careful reading and suggestions.

6 REFERENCES

[1] V. R. Basili, C. Caldiera, H.D. Rombach, *Goal Question Metric Paradigm,* in **Encyclopedia of Software Engineering**, Vol. 1, Wiley & Sons, 1994, pp 528-532.

[2] C. Cook, N. Churcher, “Modelling and Measuring Collaborative Software Engineering,” *28th Australasian Conference on Computer Science*, Newcastle, Australia, pages 267-276, January 2005.

[3] M. B. Chrissis, M. Konrad, S. Shrum, **CMMI®: Guidelines for Process Integration and Product Improvement**, 6th ed., Addison Wesley Professional, 2003.

[4] N. Jastroch, T. Marlowe, “Knowledge Transfer in Collaborative Knowledge Management: A Semiotic View”, **Journal of Systemics, Cybernetics and Informatics**, Vol. 8, No. 6, pp. 6-11, 2010.

[5] N. Jastroch, “Advancing Adaptivity in Enterprise Collaboration”, **Journal of Systemics, Cybernetics and Informatics**, Vol. 7, No. 6, pp. 7-11, 2009.

[6] N. Jastroch, “Adaptive Interenterprise Knowledge Management Systems”, *Proceedings of The 12th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI2008)*, Vol. VII, IIIS Publication, Orlando/FL, 2008.

[7] N. Jastroch, V. Kirova, C. Ku, T. Marlowe, M. Mohtashami, “Adapting Business and Technical Processes for Collaborative Software Development”, *Proceedings of the 17th International Conference on Concurrent Enterprise (ICE 2011)*, Aachen, June 2011, to appear.

[8] S. H. Kan, **Metrics and Models in Software Quality Engineering**, Second Edition, Addison Wesley, 2003.

[9] C. Kaner, W. P. Bong, “Software Engineering Metrics: What Do They Measure and How Do We Know?”, *Proceedings of the 10th International Software Metrics Symposium (METRICS 2004)*, IEEE CS Press, pages 1-12, 2004.

[10] V. Kirova, T. Marlowe, "Prendre en compte les changements dynamiques dans le développement cooperative du logiciel", **Génie Logiciel**, Decembre 2008, Numéro 87, pages 15-25.

[11] V. Kirova, N. Jastroch, C.S. Ku, T.J. Marlowe, M. Mohtashami, “Software Development Must Be Collaboration-Aware”, *Proceedings of the 22nd International Conference on Software & Systems Engineering and their Applications (ICSSEA 2010)*, Paris, December 2010.

[12] C. S. Ku, T. J. Marlowe, “Software Metrics for Collaborative Software Engineering Projects”, *Invited Session on Collaborative Knowledge Management (CKM 2010), Proceedings of the 4th International Conference on Knowledge Generation, Communication and Management (KGCM2010)*, June-July 2010.

[13] T. J. Marlowe, V. Kirova, “Addressing Change in Collaborative Software Development through Agility and Automated Traceability”, *Proceedings of The 12th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI2008)*, pp. 209–215, Orlando, USA, June-July 2008.

[14] T. J. Marlowe, V. Kirova, “High-level Component Interfaces for Collaborative Development: A Proposal**”, Journal of Systemics, Cybernetics, and Informatics**, 7 (6), pp. 1-6, 2009.

[15] T. J. Marlowe, V. Kirova, N. Jastroch, M. Mohtashami, “A Classification of Collaborative Knowledge”, *Proceedings of the 4th International Conference on Knowledge Generation, Communication and Management (KGCM2010)*, Orlando FL, July 2010.

[16] M. Mohtashami, The Antecedents and Impacts of Information Processing Effectiveness in Inter-Organizational Collaborative Software Development, Ph.D. Thesis, Rutgers University, July 2006.

[17] M. Mohtashami, T. Marlowe, V. Kirova, F. Deek, “Risk Management for Collaborative Software Development”, **Journal of Information Systems Management**, 25 (4), 20–30, Fall 2006.

[18] M. Mohtashami, T. Marlowe, V. Kirova, F. Deek, “Risk-Driven Management Contingency Policies in Collaborative Software Development”, **Intl. Journal of Information Technology and Management**, to appear, 2011.

[19] M. Mohtashami, T. Marlowe, V. Kirova, F. P. Deek, “A Comparison of Three Modes of Collaboration”, *15th Americas Conference on Information Systems (AMCIS 2009)* [CD-ROM], August 2009.

[20] M. Mohtashami, T.J. Marlowe, V. Kirova, “Lost In Translation: When Outsourcing Decouples Development*”, 41st Annual Meeting of the Design Sciences Institute (DSI 2010)* [CD-ROM], San Diego CA, November 2010.

[21] C. R. Moore, *Performance Measures for Knowledge Management*, in: Jay Liebowitz (ed.), **Knowledge Management Handbook**, CRC Press, Boca Raton, London, New York, Washington D.C., 1999.

[22] R. Reinhardt, Wissen als Ressource: theoretische Grundlagen, Methoden und Instrumente zur Erfassung von Wissen, Europ. Verlag der Wissenschaften, Frankfurt a. M., 2002. [in German only – Habilitationsschrift]

[23] R. v. Solingen, E. Berghout, **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**, McGraw Hill, London, 1999.

[24] K.E. Sveiby, **The New Organisational Wealth: Managing & Measuring Knowledge Based Assets**, Berrett-Koehler Publishers Inc., San Francisco, 1997